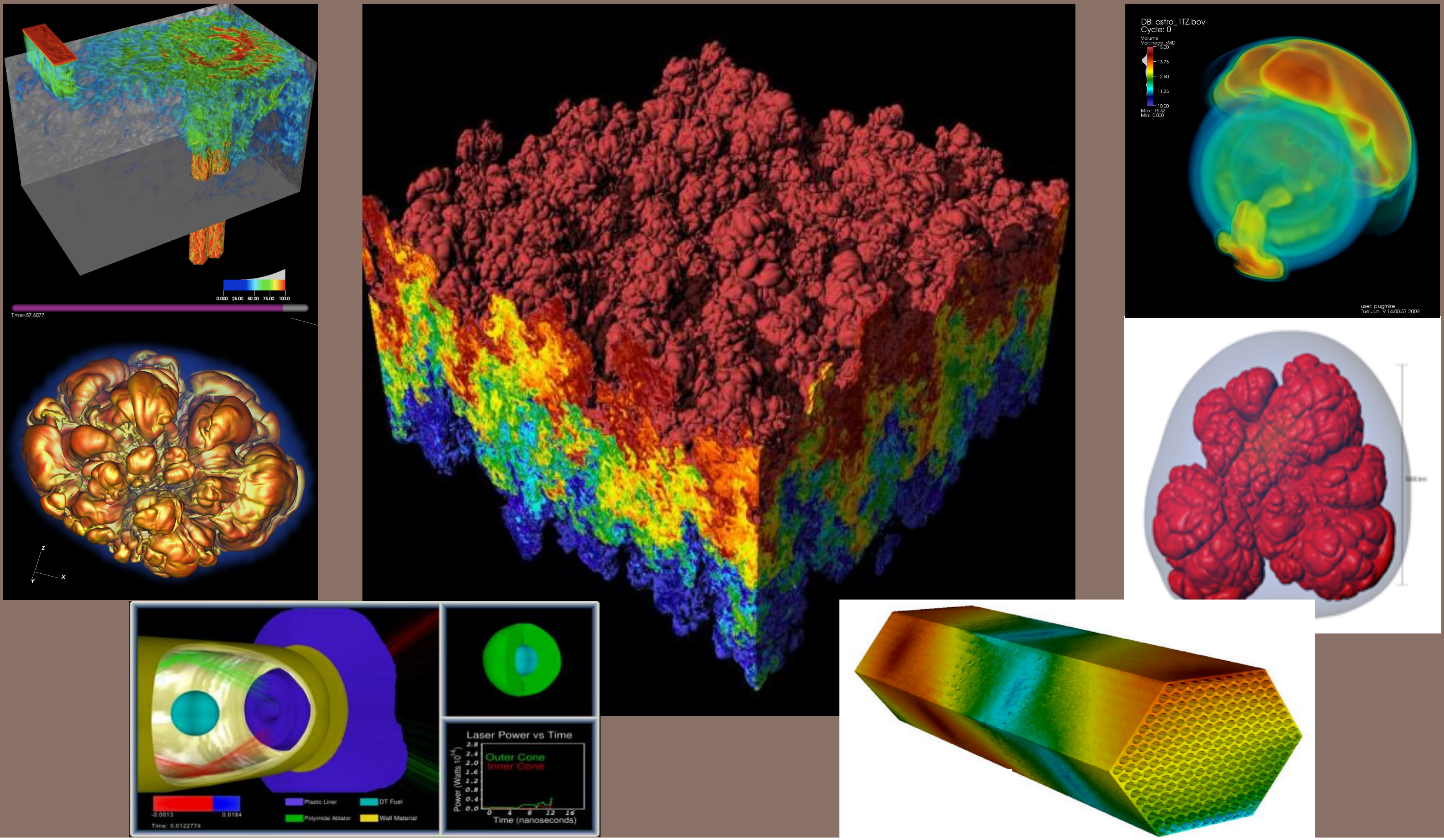
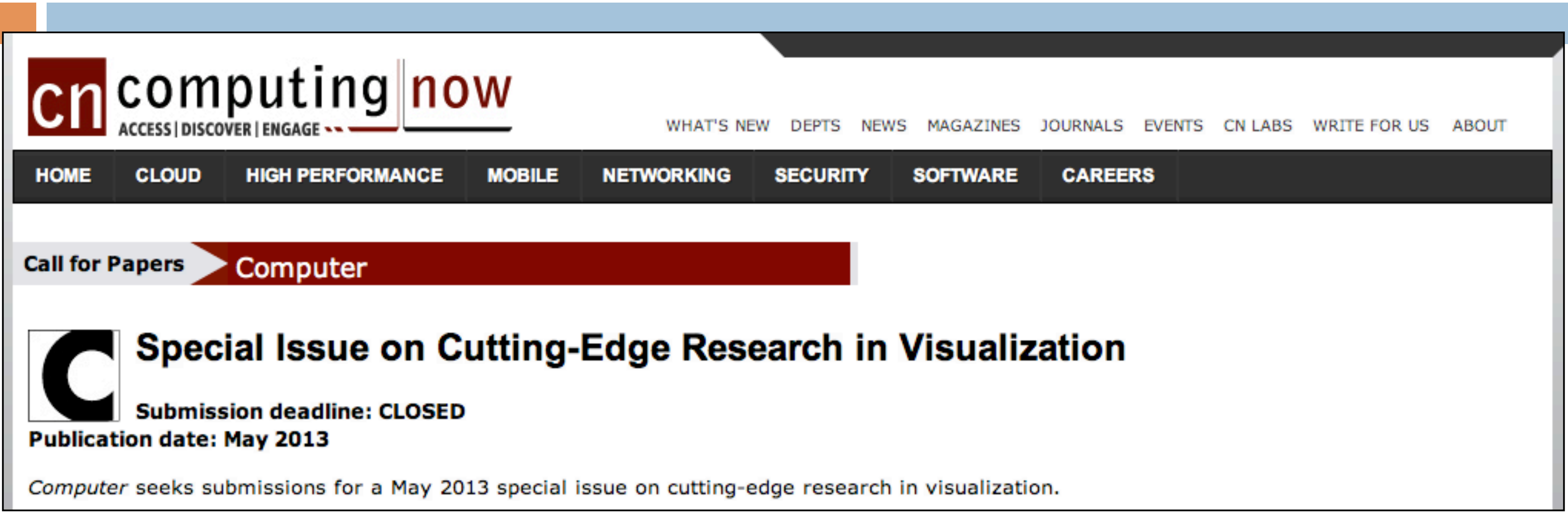
EXASCALE VISUALIZATION:
GET READY FOR A WHOLE NEW WORLD



March 26, 2013

Hank Childs, University of Oregon

big data versus Big Data



Special Issue on Cutting-Edge Research in Visualization

Submission deadline: CLOSED
Publication date: May 2013

Computer seeks submissions for a May 2013 special issue on cutting-edge research in visualization.

- (HPC) big data: large, homogeneous arrays, read from a parallel disk, and processed with symmetric resources
- Big Data: heterogeneous, unstructured data, located in a distributed setting and processed with asymmetric resources

Outline

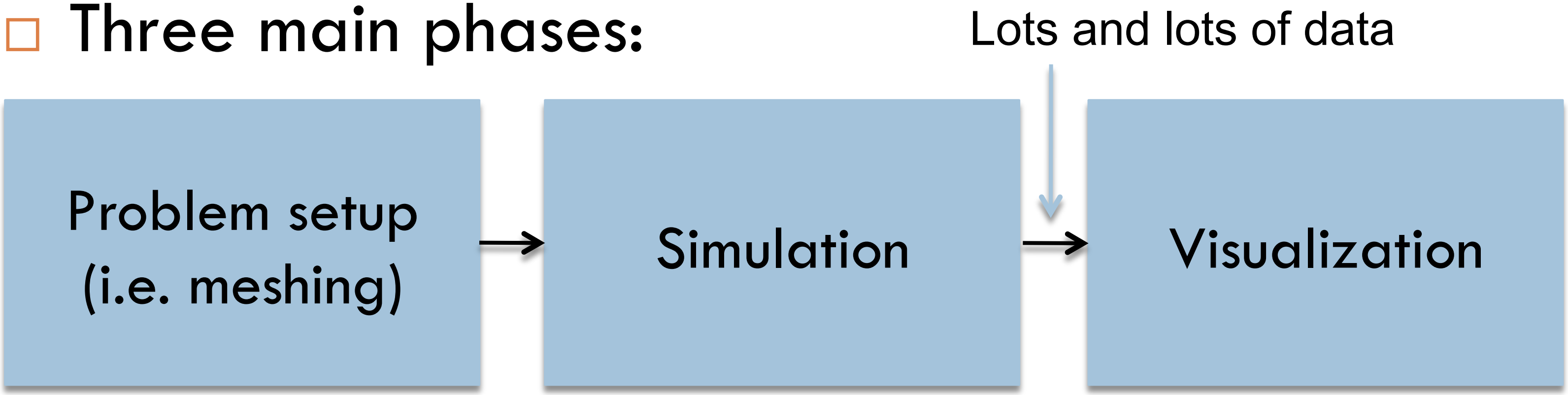
- Quick Background In
  - Scientific Visualization
  - High Performance Computing (HPC)
  - Vis+HPC
- Petascale Visualization
- Exascale Computing
- Exascale Visualization

Outline

- Quick Background In
  - Scientific Visualization
  - High Performance Computing (HPC)
  - Vis+HPC
- Petascale Visualization
- Exascale Computing
- Exascale Visualization

Visualization is a key aspect of the simulation process.

- Three main phases:

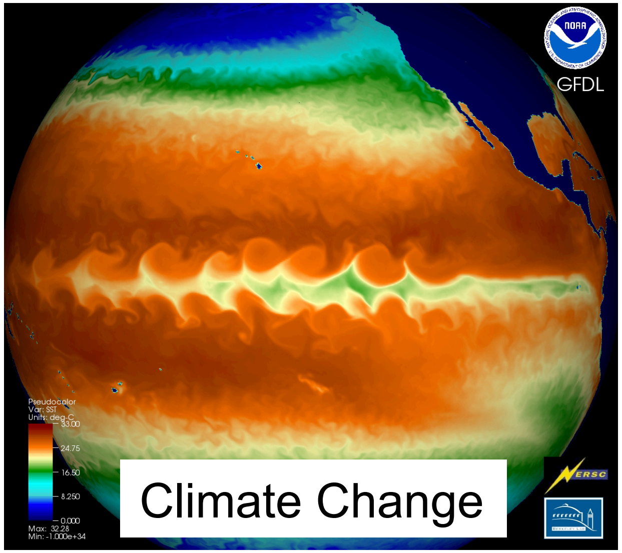


- Visualization is used primarily in three ways:
  - Scientists confirm their simulation is running correctly.
  - Scientists explore data, leading to new insights.
  - Scientists communicate simulation results to an audience.

The scientific simulation community makes heavy use of supercomputers.

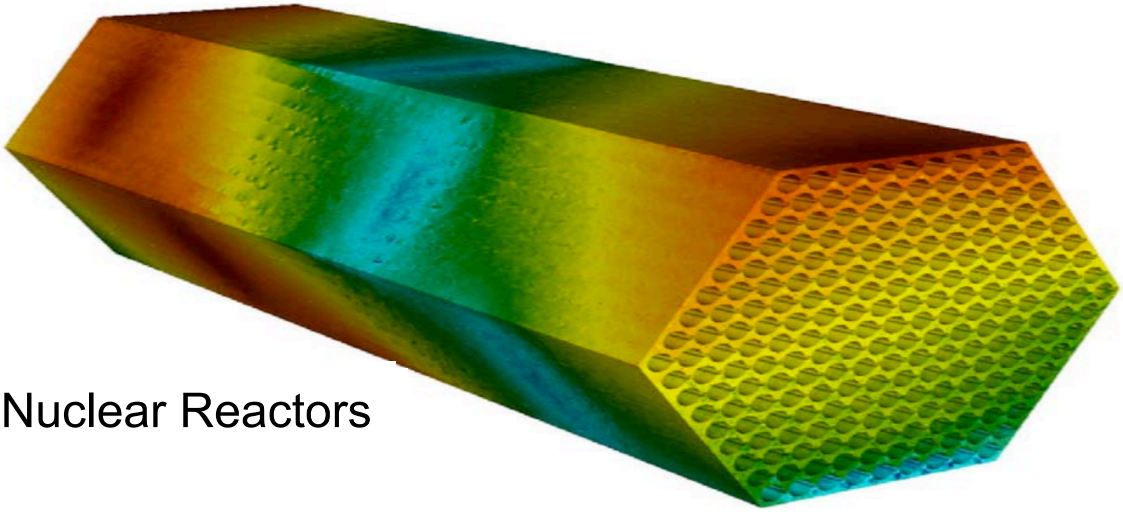
- **Why simulation?**
  - Simulations are sometimes more cost effective than experiments.

- **Why extreme scale?**
  - More compute cycles, more memory, etc, lead for faster and/or more accurate simulations.

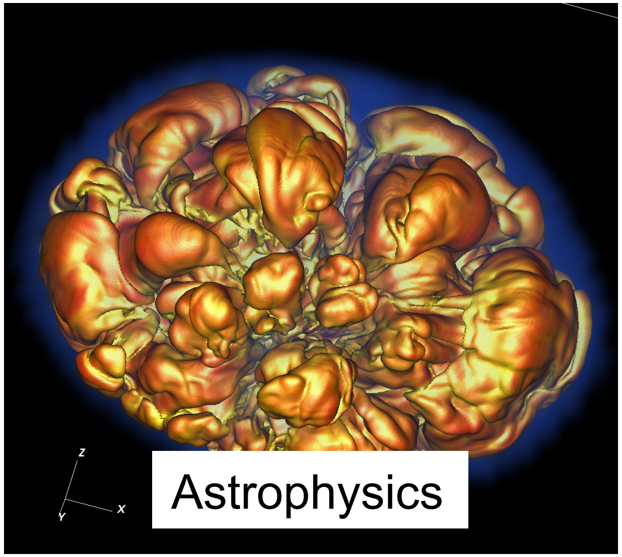


Climate Change

Image credit: Prabhat, LBNL



Nuclear Reactors



Astrophysics

The scientific simulation community makes heavy use of supercomputers.

- **How big are these machines?**
  - Measured in “FLOPs” = floating point operations per second
  - 1 GigaFLOP = 1 billion FLOPs
  - 1 TeraFLOP = 1000 GigaFLOPs
  - 1 PetaFLOP = 1,000,000 GigaFLOPs
    - → where we are today
  - 1 ExaFLOP = 1,000,000,000 GigaFLOPs
    - → potentially arriving as soon as 2018

LLNL Sequoia
#2 on Top500.org, 20 PFLOPs

ORNL Titan
#1 on Top500.org, 27 PFLOPS

RIKEN K / Kei computer
#3 on Top500.org, 10PFLOPs

The (DOE) Case for the Exascale

Scientific Grand Challenges

CHALLENGES IN CLIMATE CHANGE SCIENCE AND THE ROLE OF COMPUTING AT THE EXTREME SCALE

November 6-7, 2008 · Washington DC

Climate

Scientific Grand Challenges

FOREFRONT QUESTIONS IN NUCLEAR SCIENCE AND THE ROLE OF COMPUTING AT THE EXTREME SCALE

January 26-28, 2009 • Washington, D.C.

Nuclear Physics

Scientific Grand Challenges

FUSION ENERGY SCIENCES AND THE ROLE OF COMPUTING AT THE EXTREME SCALE

March 18-20, 2009 · Washington, DC

Fusion

Scientific Grand Challenges

Science Based Nuclear Energy Systems Enabled by Advanced Modeling and Simulation at the Extreme Scale

May 11-12, 2009 · Washington DC

Nuclear Reactors

Scientific Grand Challenges

CHALLENGES FOR UNDERSTANDING THE QUANTUM UNIVERSE AND THE ROLE OF COMPUTING AT THE EXTREME SCALE

December 9-11, 2008 · Menlo Park, CA

High Energy Physics

Scientific Grand Challenges

DISCOVERY IN BASIC ENERGY SCIENCES THE ROLE OF COMPUTING AT THE EXTREME SCALE

August 13-15, 2009 · Washington, DC

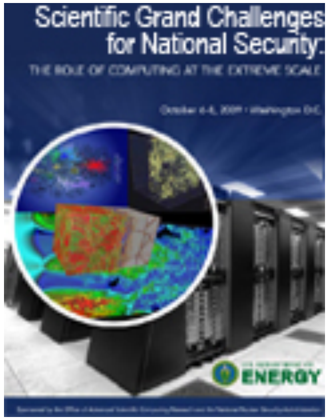
Material Science & Chemistry

Scientific Grand Challenges

Opportunities in Biology at the Extreme Scale of Computing

August 17-19, 2009 | Chicago, IL

Biology



Scientific Grand Challenges for National Security:

THE ROLE OF COMPUTING AT THE EXTREME SCALE

October 6-8, 2009 · Washington D.C.

National Security

International Exascale Software Project

www.exascale.org

INTERNATIONAL EXASCALE SOFTWARE PROJECT ROADMAP 1.0

Jack Dongarra
Pete Beckman
Terry Moore
Patrick Aerts
Giovanni Aloisio
Jean-Claude Andre
David Barkai
Jean-Yves Berthou
Taisuke Boku
Bertrand Braunschweig
Franck Cappello
Barbara Chapman
Xuebin Chi

Alok Choudhary
Sudip Dosanjh
Thom Dunning
Sandro Fiore
Al Geist
Bill Gropp
Robert Harrison
Mark Hereld
Michael Heroux
Adolfy Hoisie
Koh Hotta
Yutaka Ishikawa
Fred Johnson

Sanjay Kale
Richard Kenway
David Keyes
Bill Kramer
Jesus Labarta
Alain Lichnewsky
Thomas Lippert
Bob Lucas
Barney Maccabe
Satoshi Matsuoka
Paul Messina
Peter Michielse
Bernd Mohr

Matthias Mueller
Wolfgang Nagel
Hiroshi Nakashima
Michael E. Papka
Dan Reed
Mitsuhisa Sato
Ed Seidel
John Shalf
David Skinner
Marc Snir
Thomas Sterling
Rick Stevens
Fred Streitz

Bob Sugar
Shinji Sumimoto
William Tang
John Taylor
Rajeev Thakur
Anne Trefethen
Mateo Valero
Aad van der Steen
Jeffrey Vetter
Peg Williams
Robert Wisniewski
Kathy Yelick

SPONSORS

Office of Science U.S. Department of Energy
NSF
ANR
cea
CERFACS
CRAY THE SUPERCOMPUTER COMPANY
eDF
EPSRC Engineering and Physical Sciences Research Council
FUJITSU
INRIA
KYOTO UNIVERSITY FOUNDED 1897
GENCI
NVIDIA
RIKEN
東京大学 THE UNIVERSITY OF TOKYO
筑波大学

The International Exascale Software Roadmap,
J. Dongarra, P. Beckman, et al.,
International Journal of High Performance Computer Applications **25**(1), 2011, ISSN 1094-3420. (Publ. 6 Jan 2011)

Outline

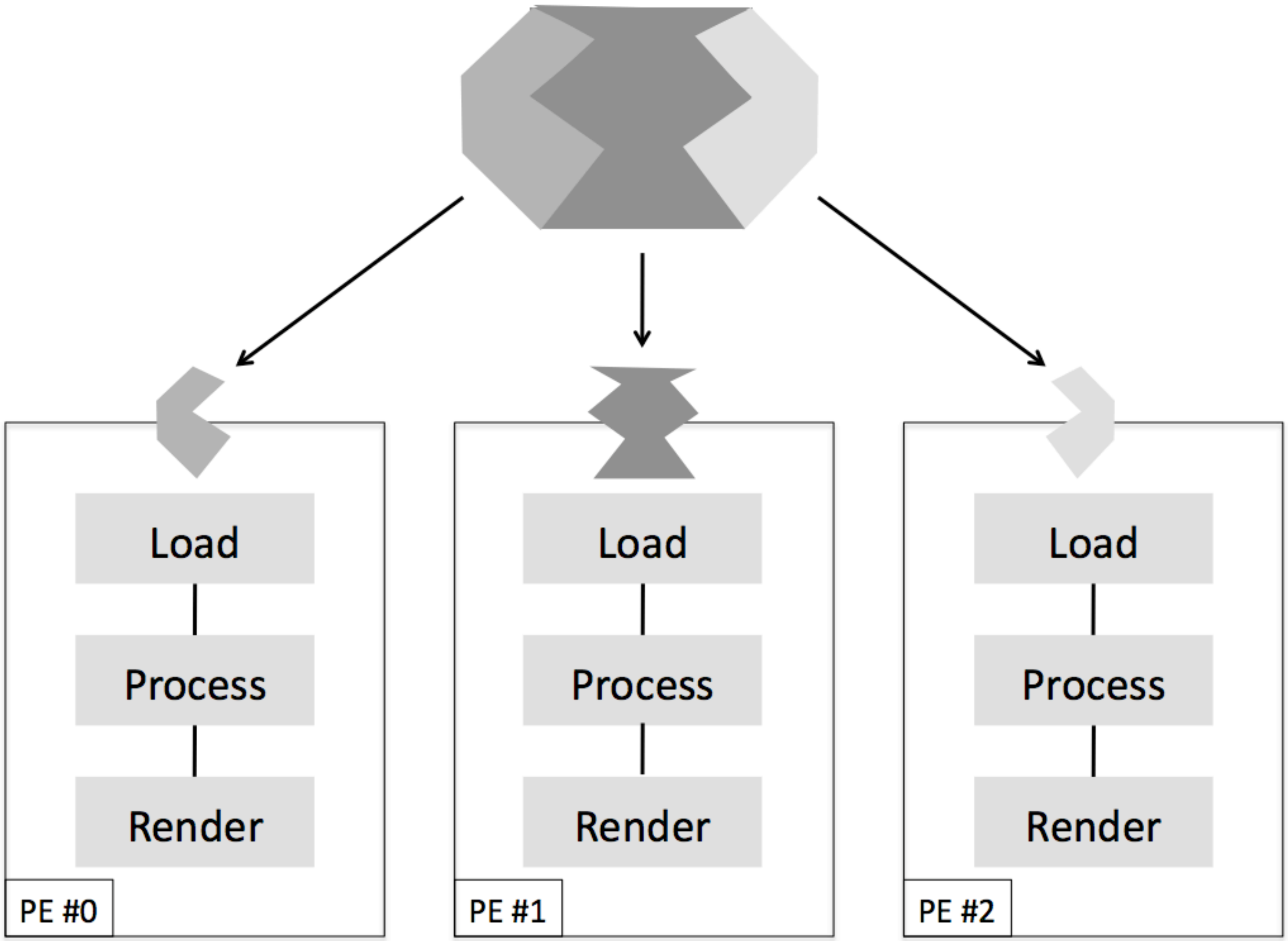
- Quick Background In
  - Scientific Visualization
  - High Performance Computing
  - Vis+HPC
- Petascale Visualization
- Exascale Computing
- Exascale Visualization

Defining "big data" for visualization

- Big data: data that is too big to be processed in its entirety all at one time because it exceeds the available memory.

Criterion	Approaches
In its entirety	Data subsetting / multi-resolution
All at one time	Streaming (e.g. out of core)
Exceeds available memory	Parallelism

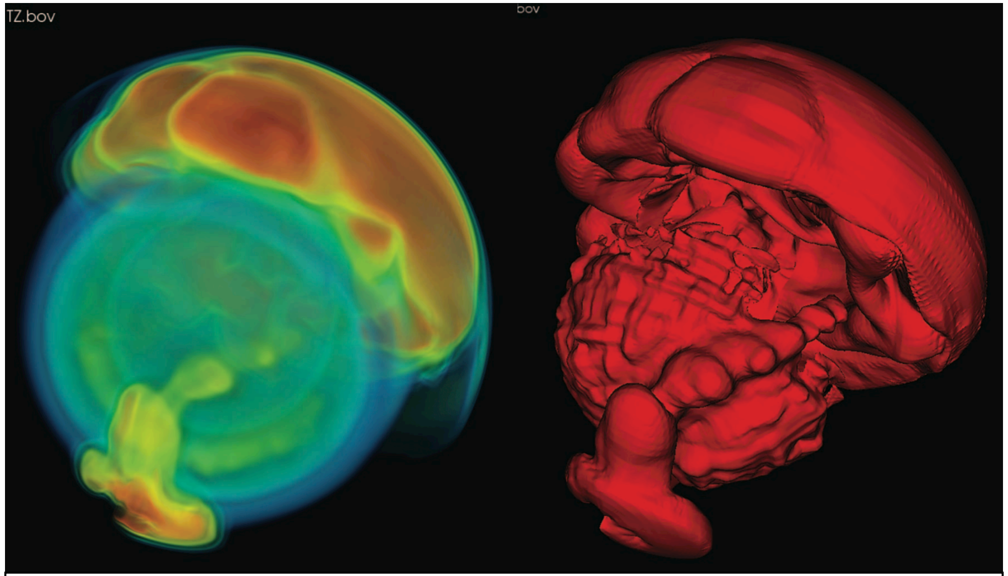
Data parallelism is the dominant paradigm for processing.



PE = Processing Element

How far can data parallelism go?

- Study: scale the data parallel approach to ***trillions of cells*** and tens of thousands of cores, varying supercomputing environment, I/O pattern, and data set.
- Finding: the approach works well ***for some algorithms***, but ***I/O is a limiting factor***.



Volume rendering and isosurface of 1 trillion cell astrophysics data set, using 16,000 cores of LBNL Franklin machine.

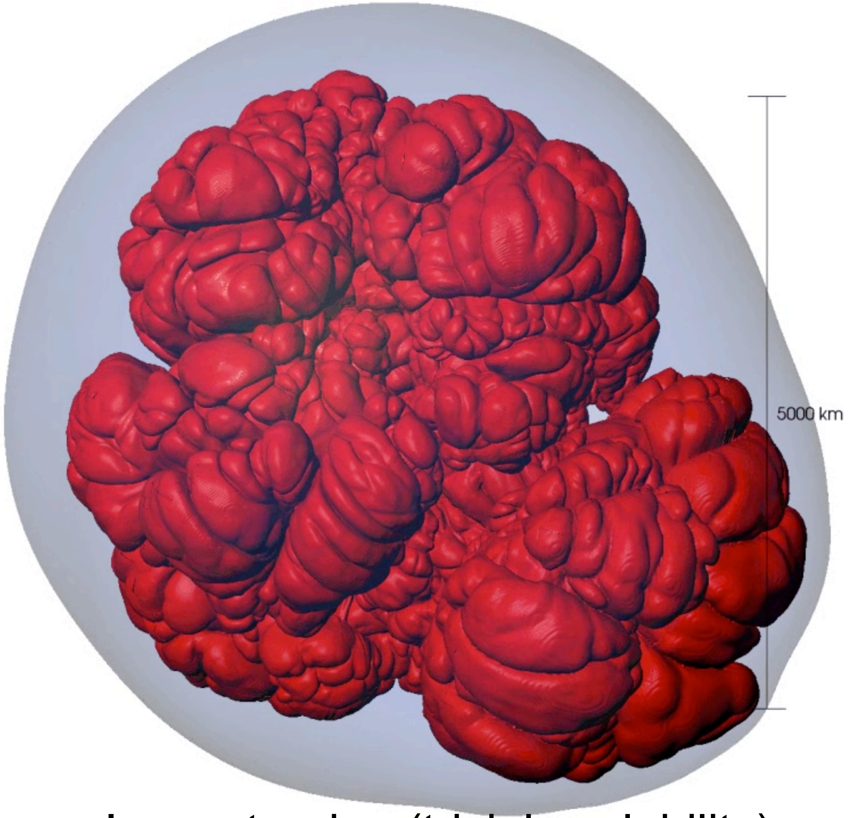
H. Childs, D. Pugmire, S. Ahern, B. Whitlock, M. Howison, Prabhat, G. Weber, and E. W. Bethel. "Extreme Scaling of Production Visualization Software on Diverse Architectures", Computer Graphics and Applications, volume 30, number 3, pp. 22-31, May/June 2010.

Outline

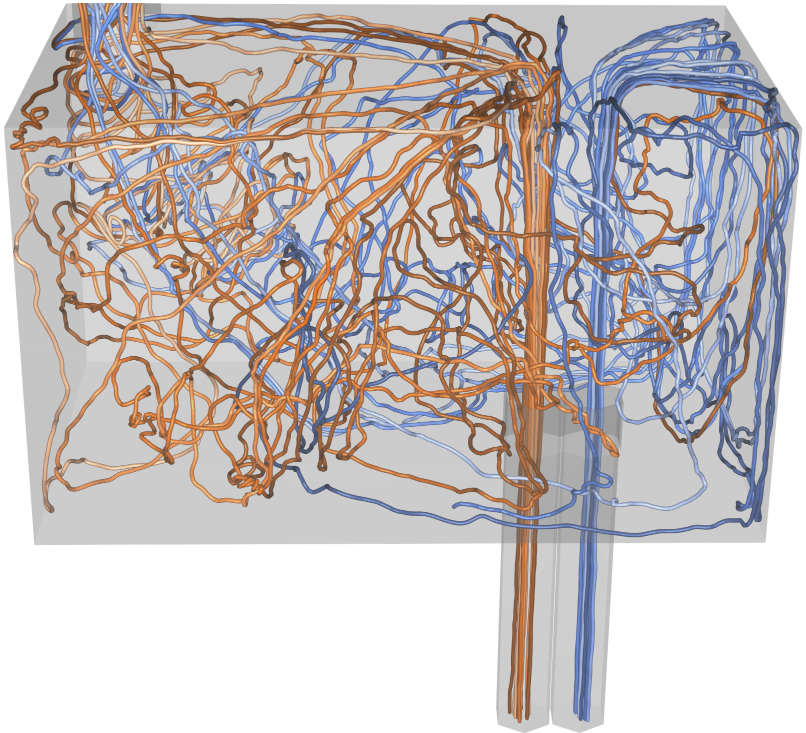
- Quick Background In
  - Scientific Visualization
  - High Performance Computing
  - Vis+HPC
- Petascale Visualization
- Exascale Computing
- Exascale Visualization

The two scale challenges for petascale visualization

- Scalable algorithms
- Minimize I/O



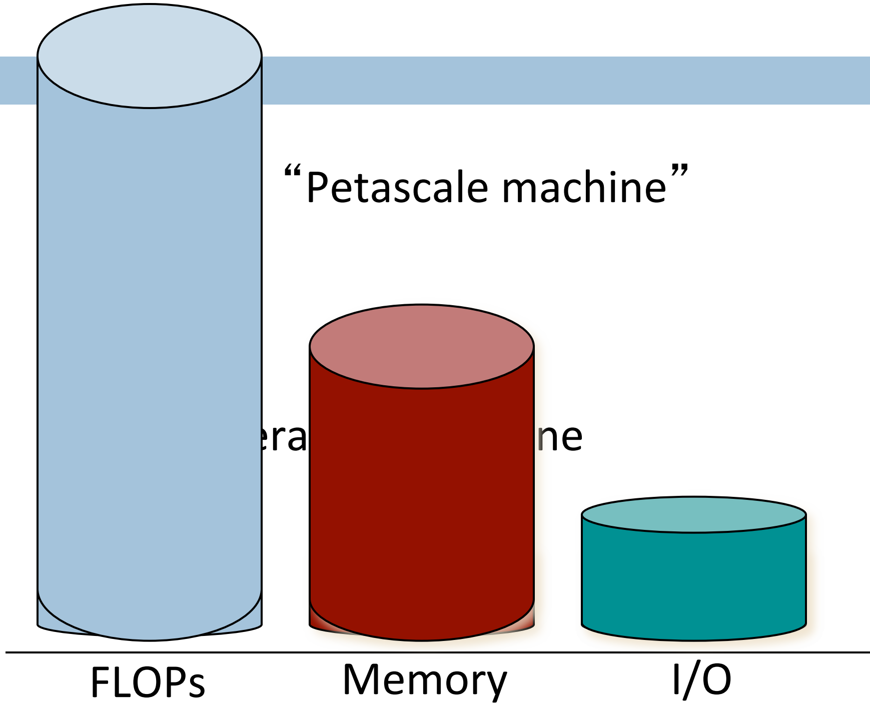
Isocontouring (trivial scalability)



Streamlines (difficult scalability)

I/O and visualization

- Data parallelism (for visualization) is almost always >50% I/O and sometimes 98% I/O
- Amount of data to visualize is typically O(total mem)
- Two big factors:
  ① how much data you have to read
  ② how fast you can read it
- → Relative I/O (ratio of total memory and I/O) is key



Trends in I/O

Machine	Year	Time to write memory
ASCI Red	1997	300 sec
ASCI Blue Pacific	1998	400 sec
ASCI White	2001	660 sec
ASCI Red Storm	2004	660 sec
ASCI Purple	2005	500 sec
Jaguar XT4	2007	1400 sec
Roadrunner	2008	1600 sec
Jaguar XT5	2008	1250 sec

c/o David Pugmire, ORNL

Why is relative I/O getting slower?

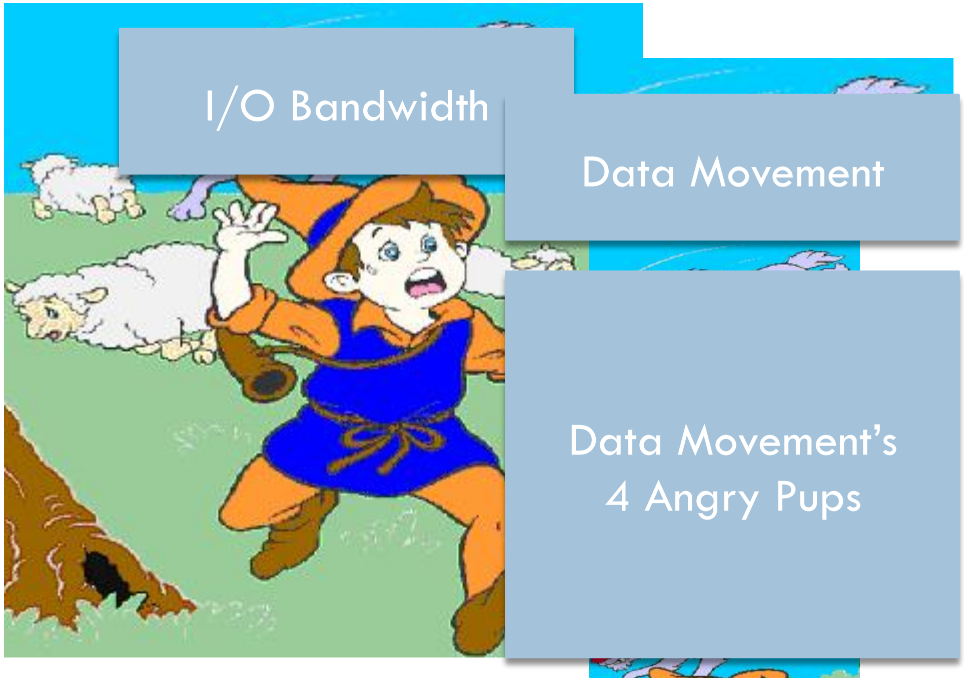
- I/O is quickly becoming a dominant cost in the overall supercomputer procurement.
  - And I/O doesn’t pay the bills.
- Simulation codes aren’t as exposed.

We need to de-emphasize I/O in our visualization and analysis techniques.

The message from this talk…

I/O Bandwidth

Petascale Visualization



I/O Bandwidth

Data Movement

Data Movement's
4 Angry Pups

Exascale Visualization

In situ processing is a solution for both of these problems.

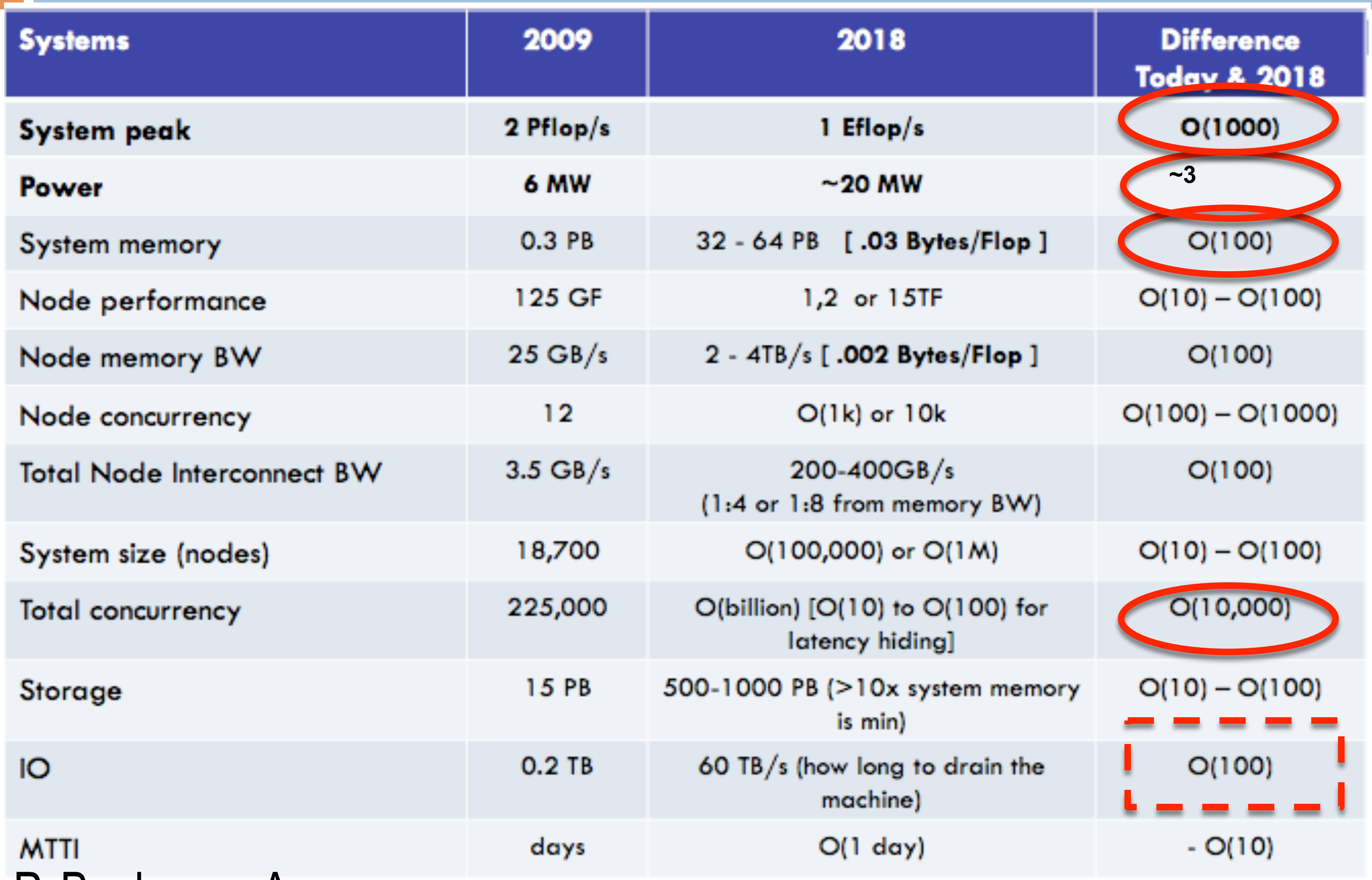
In Situ Processing

- Defined: couple visualization and analysis routines with the simulation code (no I/O)
- Pros:
  - No I/O!
  - Can access all the data
  - Computational power readily available
- Cons:
  - Must know what you want to look for a priori
  - Increasing complexity
  - Constraints (memory, network)

Outline

- Quick Background In
  - Scientific Visualization
  - High Performance Computing
  - Vis+HPC
- Petascale Visualization
- Exascale Computing
- Exascale Visualization

Exascale: a heterogeneous, distributed memory *GigaHz KiloCore MegaNode* system



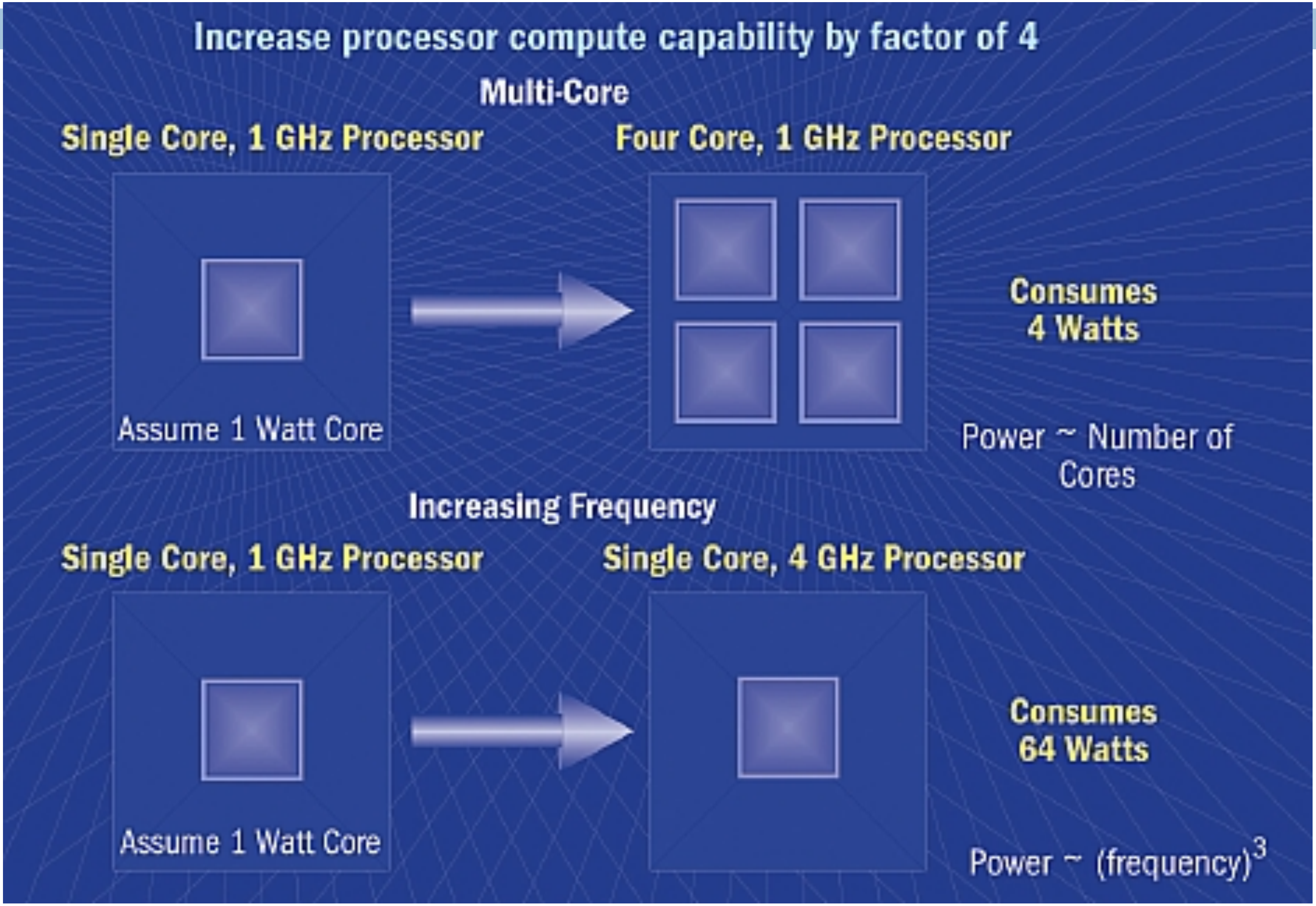
Systems	2009	2018	Difference Today & 2018
System peak	**2 Pflop/s**	**1 Eflop/s**	**O(1000)**
Power	**6 MW**	**~20 MW**	**~3**
System memory	0.3 PB	32 - 64 PB [**.03 Bytes/Flop**]	O(100)
Node performance	125 GF	1,2 or 15TF	O(10) – O(100)
Node memory BW	25 GB/s	2 - 4TB/s [**.002 Bytes/Flop**]	O(100)
Node concurrency	12	O(1k) or 10k	O(100) – O(1000)
Total Node Interconnect BW	3.5 GB/s	200-400GB/s (1:4 or 1:8 from memory BW)	O(100)
System size (nodes)	18,700	O(100,000) or O(1M)	O(10) – O(100)
Total concurrency	225,000	O(billion) [O(10) to O(100) for latency hiding]	O(10,000)
Storage	15 PB	500-1000 PB (>10x system memory is min)	O(10) – O(100)
IO	0.2 TB	60 TB/s (how long to drain the machine)	O(100)
MTTI	days	O(1 day)	- O(10)

c/o P. Beckman, Argonne

Exascale assumptions

- The machine will be capable of one exaflop.
- The machine will cost < $200M.
- The machine will use < 20MW.
- The machine may arrive as early as 2018.

Hurdle #1: power requires slower clocks and greater concurrency



Increase processor compute capability by factor of 4

Multi-Core

Single Core, 1 GHz Processor

Four Core, 1 GHz Processor

Assume 1 Watt Core

Consumes 4 Watts

Power ~ Number of Cores

Increasing Frequency

Single Core, 1 GHz Processor

Single Core, 4 GHz Processor

Assume 1 Watt Core

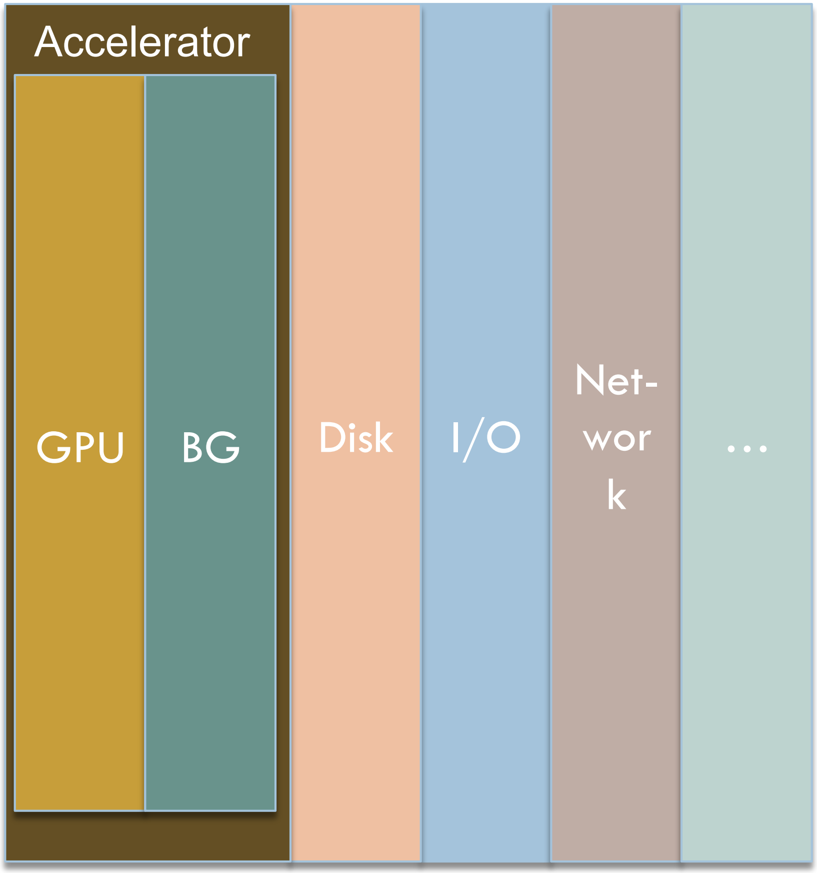
Consumes 64 Watts

Power ~ $(\text{frequency})^3$

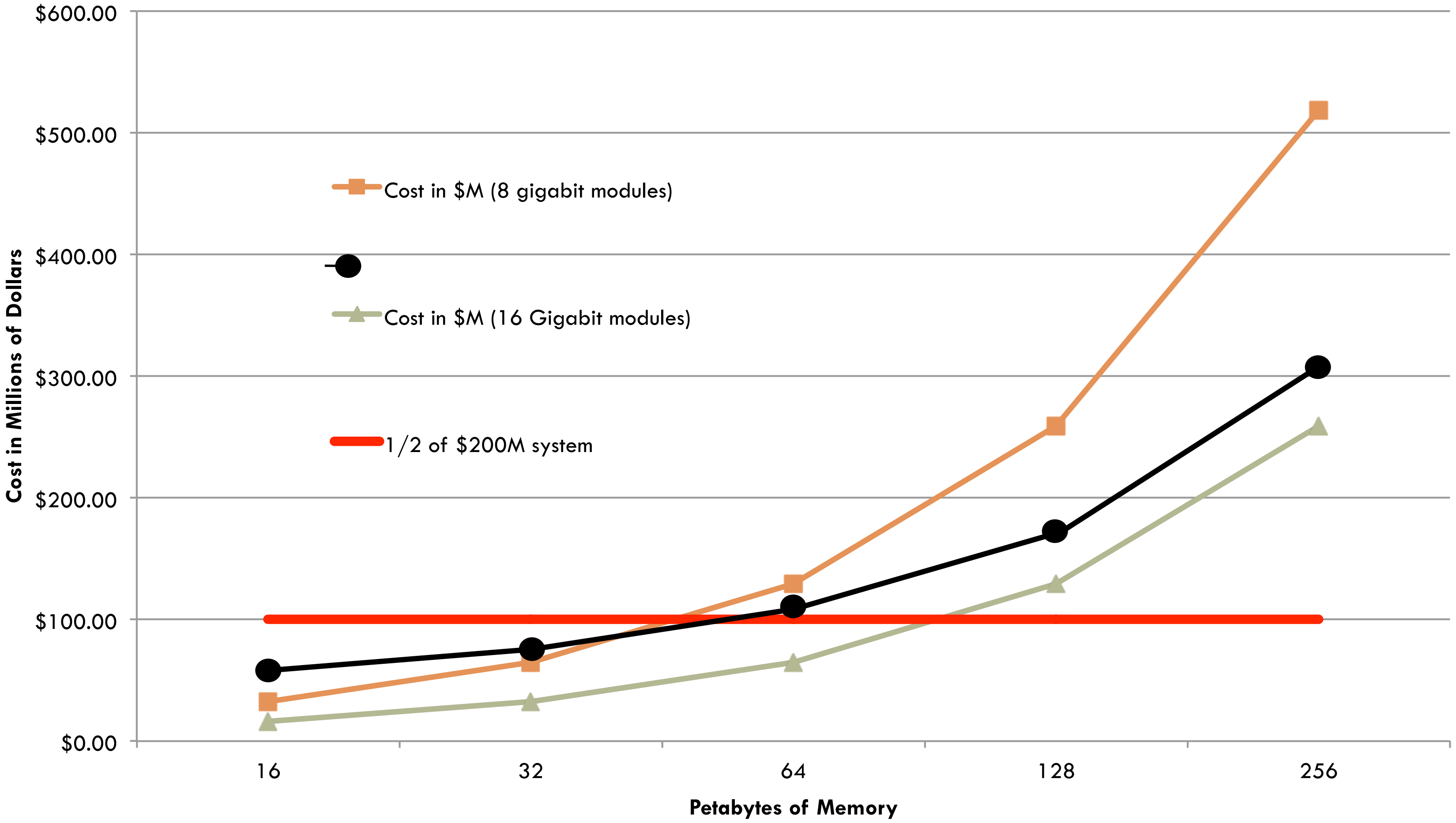
c/o *SciDAC Review* 16, February 2010

Accelerator technologies

- Currently simultaneously thinking about two different accelerator technologies:
  - IBM BlueGene's successor – some architectural merger of BlueGene, Power, and Cell
  - GPU / GPU evolution
- Referred to as "swim lanes": a visual element used in process flow diagrams, or flowcharts, that visually distinguishes responsibilities for sub-processes of a business process.

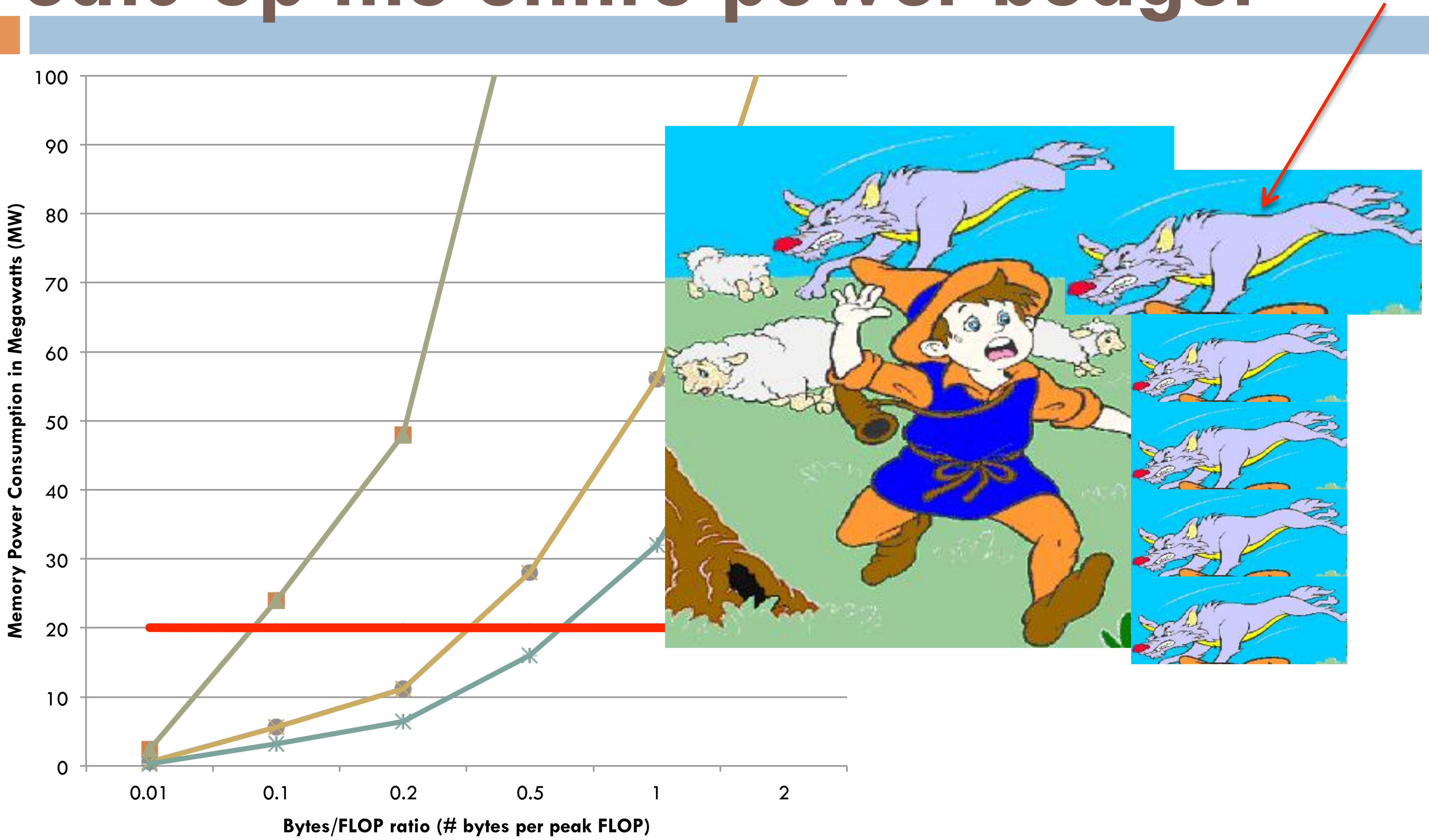


Hurdle #2: memory capacity eats up the entire fiscal budget



c/o John Shalf, LBNL

Hurdle #3: memory bandwidth eats up the entire power budget



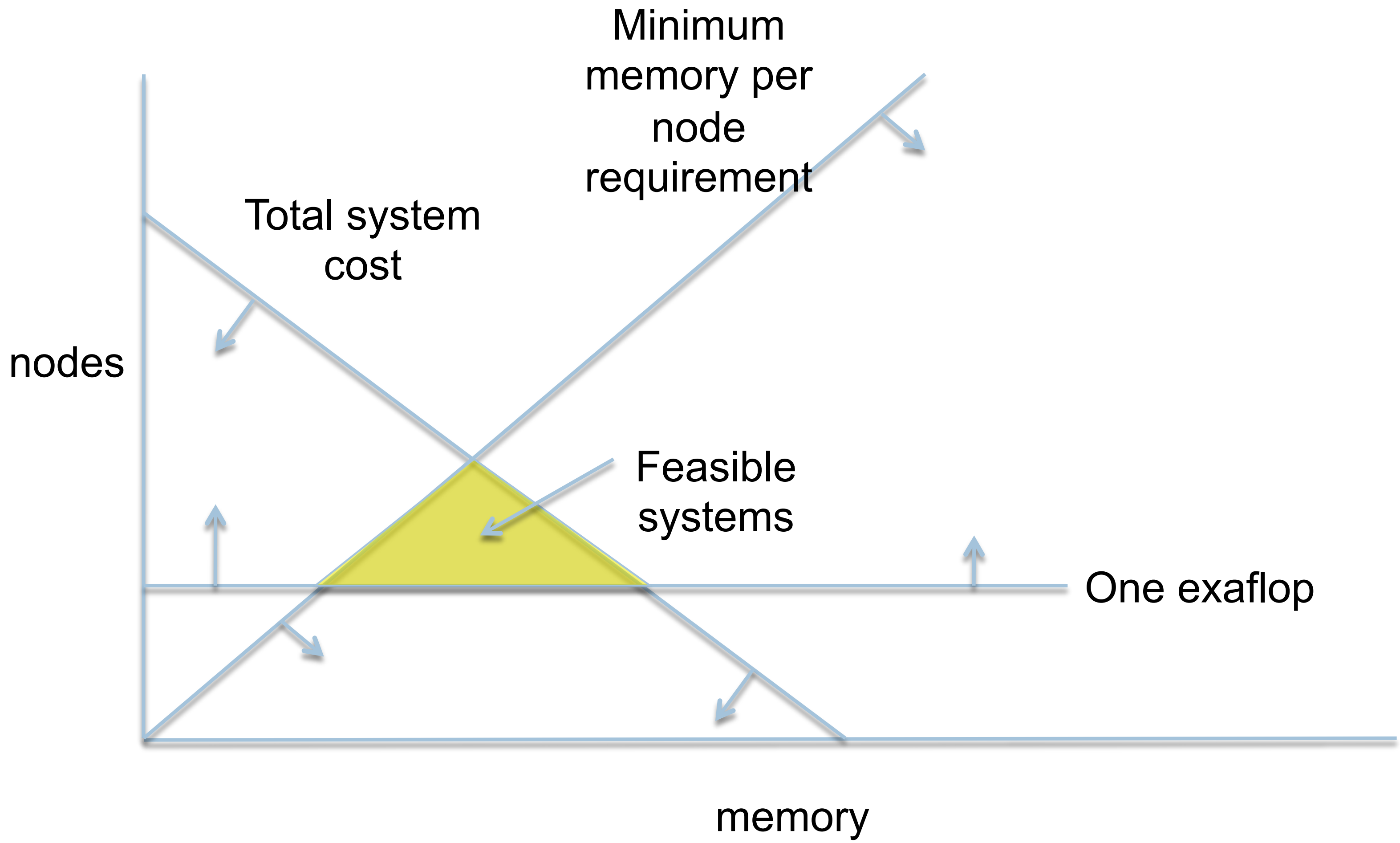
c/o John Shalf, LBNL

The change in memory bandwidth to compute ratio will lead to new approaches.

- Example: linear solvers
  - They start with a rough approximation and converge through an iterative process.
    - 1.125 → 1.1251 → 1.125087 → 1.12508365
  - Each iteration requires sending some numbers to neighboring processors to account for neighborhoods split over multiple nodes.
  - Proposed exascale technique: devote some threads of the accelerator to calculating the difference from the previous iteration and just sending the difference.
    - Takes advantage of "free" compute and minimizes expensive memory movement.

Inspired by David Keyes, KAUST and Richard Bowers, BU

The trade space for exascale is very complex.



c/o A. White, LANL

Architectural changes will make writing fast and reading slow.

- Great idea: put SSDs on the node
  - Great idea for the simulations …
  - … scary world for visualization and analysis
    - We have lost our biggest ally in lobbying the HPC procurement folks
    - We are unique as data consumers
- $200M is not enough…
  - The quote: "1/3 memory, 1/3 I/O, 1/3 networking … and the flops are free"
  - Budget stretched to its limit and won't spend more on I/O.

Architectural changes will make writing fast and reading slow.

- Great idea: put SSDs on the node
  - Great idea for the simulations …
  - … scary world for visualization and analysis
    - We have lost our biggest ally in lobbying the HPC procurement folks
    - We are unique as data consumers
- $200M is not enough…
  - The quote: "1/3 memory, 1/3 I/O, 1/3 networking … and the flops are free"
  - Budget stretched to its limit and won't spend more on I/O.

Summary of Exascale Challenges

- The hardware architecture will be different than the petascale.
  - Not just multi-core, but many-core
- Achieving an ExaFLOP with $200M and 20MW budgets requires complex tradeoffs.
- Data movement will be a key issue for exascale visualization.
  - End of traditional post-processing?
  - Even movement around the machine will be hard.

Outline

- Quick Background In
  - Scientific Visualization
  - High Performance Computing
  - Vis+HPC
- Petascale Visualization
- Exascale Computing
- Exascale Visualization

Summarizing exascale visualization

- Hard to get data off the machine.
  - And we can't read it in if we do get it off.
  - Hard to even move it around the machine.

- → We must find ways to visualize & analyze data without doing so much I/O
- Multiresolution techniques: compelling
- In situ techniques: the focal point

4 Angry Pups

- In Situ Systems Research
- Programming Languages
- Memory Footprint
- Exploration at the Exascale

4 Angry Pups

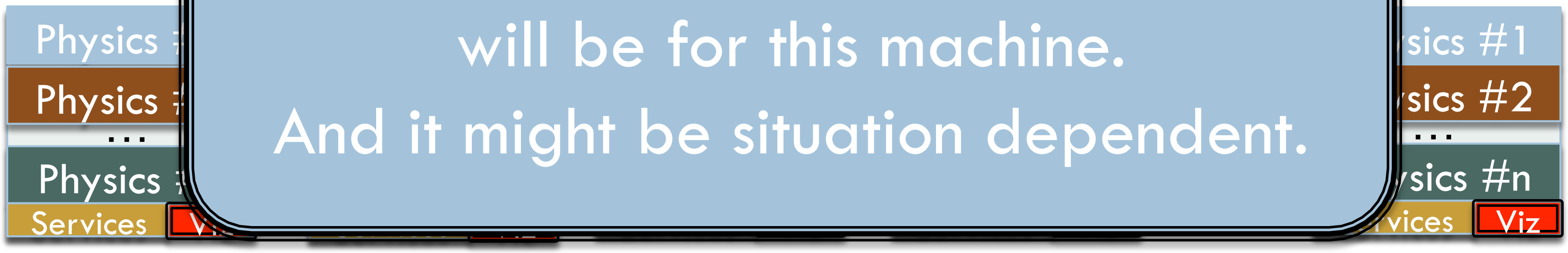
- In Situ Systems Research
- Programming Languages
- Memory Footprint
- Exploration at the Exascale

Summarizing flavors of in situ

In Situ Technique	Aliases	Description	Negative Aspects
Tightly coupled	Synchronous, co-processing	Visualization and analysis have direct access to memory of simulation code	1) Very memory constrained 2) Large potential impact (performance, crashes)
Loosely coupled	Asynchronous, concurrent	Visualization and analysis run on concurrent resources and access data over network	1) Data movement costs 2) Requires separate resources
Hybrid		Data is reduced in a tightly coupled setting and sent to a concurrent resource	1) Complex 2) Shares negative aspects (to a lesser extent) of others

Possible in situ visualization scenarios

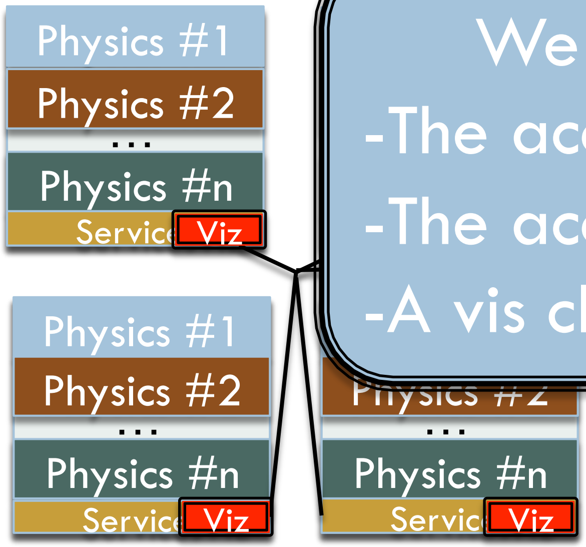
Visualization



We don't know what the best technique will be for this machine.

And it might be situation dependent.

… or visualization could be done on a separate node located nearby dedicated to visualization/analysis/IO/etc. (loosely coupled)



We will possibly need to run on:

-The accelerator in a lightweight way

-The accelerator in a heavyweight way

-A vis cluster (?)

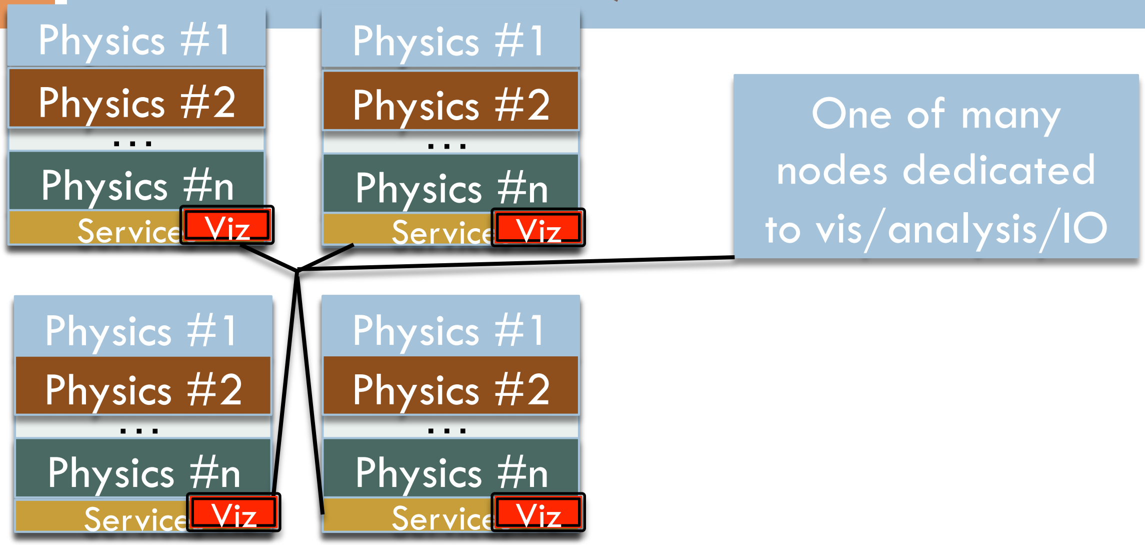
a high memory quad-core running Linux!

to HW on rest of exascale machine (e.g. GPU)

… And likely many more configurations

the data is reduced and sent to dedicated resources off machine!

Reducing data to results (e.g. pixels or numbers) can be hard.



- Must to reduce data every step of the way.
  - Example: contour + normals + render
    - Important that you have less data in pixels than you had in cells. (*)
    - Could contouring and sending triangles be a better alternative?
  - Easier example: synthetic diagnostics

4 Angry Pups

□ In Situ Systems Research

□ Programming Languages

□ Memory Footprint

□ Exploration at the Exascale

Angry Pup #2: Programming Language

- VTK: enables the community to develop diverse algorithms for diverse execution models for diverse data models
  - Impor
  - Subst
- We nee
  - Will also be a substantial investment
- Must be:
  - Lightweight
  - Efficient
  - Able to run in a many core environment

OK, what language is this in?
OpenCL? DSL?
… not even clear how to start

Message-passing remains important at the exascale, but we lose its universality

MPI will be combined with other paradigms within a shared memory node (OpenMP, OpenCL, CUDA, etc.)

Codes will not be hardware-universal again, until a lengthy evolutionary period passes

c/o David Keyes, KAUST

4 Angry Pups

- □ In Situ Systems Research
- □ Programming Languages
- □ Memory Footprint
- □ Exploration at the Exascale

Memory efficiency

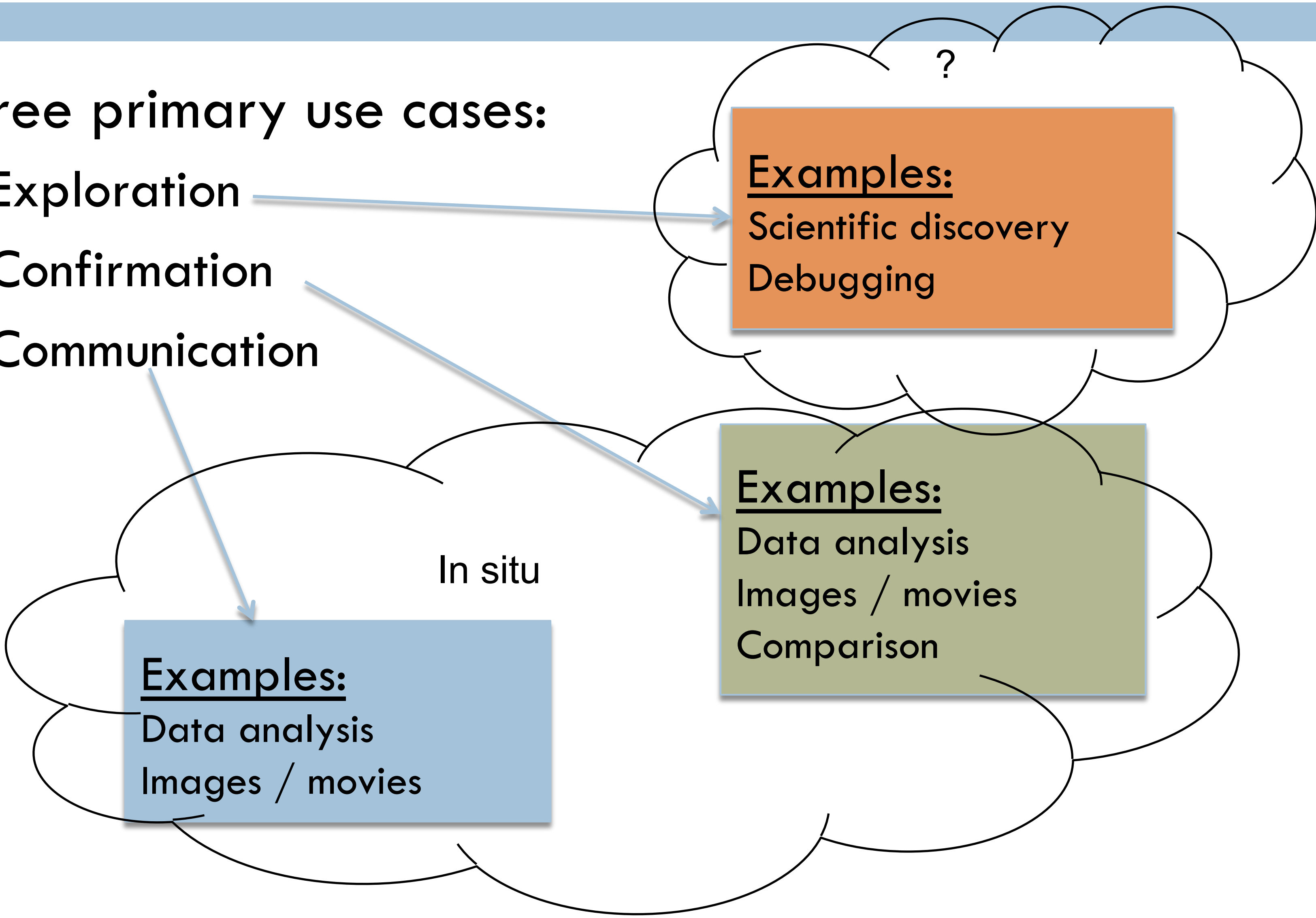
- 64 PB of memory for 1 billion cores means 64MB per core
  - (May be 10 billion cores and 6.4MB per core)
- Memory will be the 2nd most precious resource on the machine.
  - There won't be a lot left over for visualization and analysis.
- Zero copy in situ is an obvious start
  - Templates? Virtual functions?
- Ensure fixed limits for memory footprints (Streaming?)

4 Angry Pups

- □ In Situ Systems Research
- □ Programming Languages
- □ Memory Footprint
- □ Exploration at the Exascale

Do we have our use cases covered?

- Three primary use cases:
  - Exploration
  - Confirmation
  - Communication



?

Examples:

Scientific discovery

Debugging

Examples:

Data analysis

Images / movies

Comparison

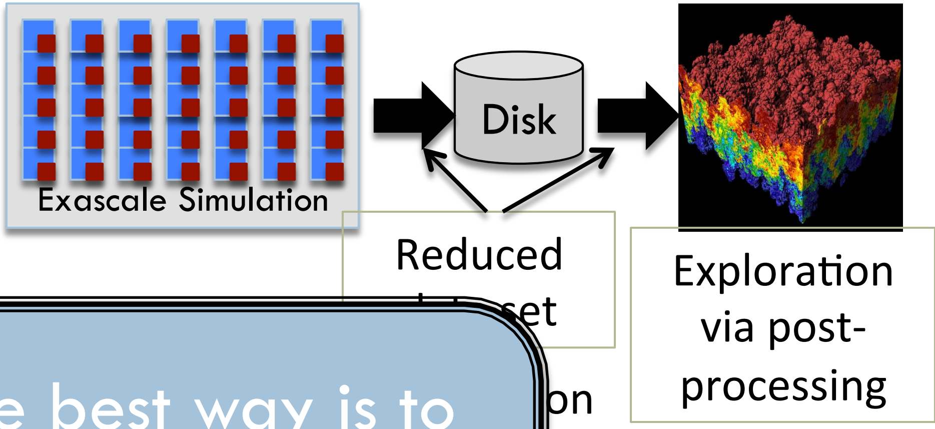
In situ

Examples:

Data analysis

Images / movies

Enabling exploration via in situ processing



- Requirement: must transform the data in a way that both reduces and enables mean

on

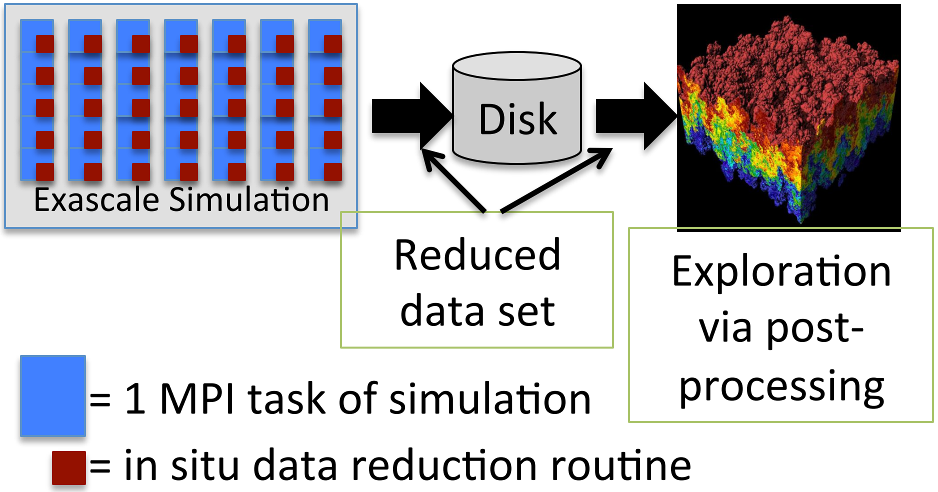
routine

It is not clear what the best way is to use in situ processing to enable exploration with post-processing … it is only clear that we need to do it.

- Subse
  - Exe
    query-driven visualization
    - User applies repeated queries to better understand data
    - New model: produce set of subsets in situ, explore it with postprocessing
  - ooks at coarse data, but can dive down to original data.
  - New model: branches of the multi-res tree are pruned if they are very similar. (compression!)

CAREER: Data Exploration at the Exascale

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY



Reduced data set

Exploration via post-processing

🟦 = 1 MPI task of simulation

🟥 = in situ data reduction routine

Novel Ideas

•*In situ processing* is viewed as a key technique for exascale computing, since it saves power by minimizing data movement. It typically assumes tasks are identified a priori.

•*Data exploration* is a labor intensive process where analysts dynamically identify questions as they explore. *It is frequently how new science is discovered.*

•In situ processing and data exploration are typically viewed as incongruent.

•We are seeking in situ reductions and transformations that will enable subsequent data exploration.

Impact and Champions

IMPACT. We will build a catalog of techniques and their efficacy (both in performance and data integrity) that will allow exascale scientists to choose the best technique for their simulation. This catalog will inform the following questions:

(1)How much data reduction with specific techniques? What are the power costs?

(2)How can these techniques be carried out at billion way concurrency?

(3)How can we create confidence in the results? How can we quantify data integrity? How can we communicate it?

Principal Investigator(s): Hank Childs, Lawrence Berkeley

Milestones/Dates/Status

This project was funded in July 2012, by the DOE Early Career program. Milestones in the early years research reduction techniques and their efficacies and in the late years develop a "cookbook" for exascale scientists.

Period	Milestone
• 7/12-6/13	Develop full evaluation of single example
• 7/13-6/14	Evaluation of second example
• 7/14-6/15	New uncertainty visualization techniques
• 7/15-6/16	Cross-product study of 400 examples
• 7/16-6/17	Develop "exascale cookbook" including insights distilled from experiments

U.S. DEPARTMENT OF ENERGY | Office of Science

Under-represented topics in this talk.

- We will have quintillions of data points … how do we meaningfully represent that with millions of pixels?
- Data is going to be different at the exascale: ensembles, multi-physics, etc.
  - The outputs of visualization software will be different.
- Nodes on exascale machine are likely not to have cache coherency
  - How well do our algorithms work in a GPU-type setting?
  - We have a huge investment in CPU-SW. What now?
- What do we have to do to support resiliency issue?

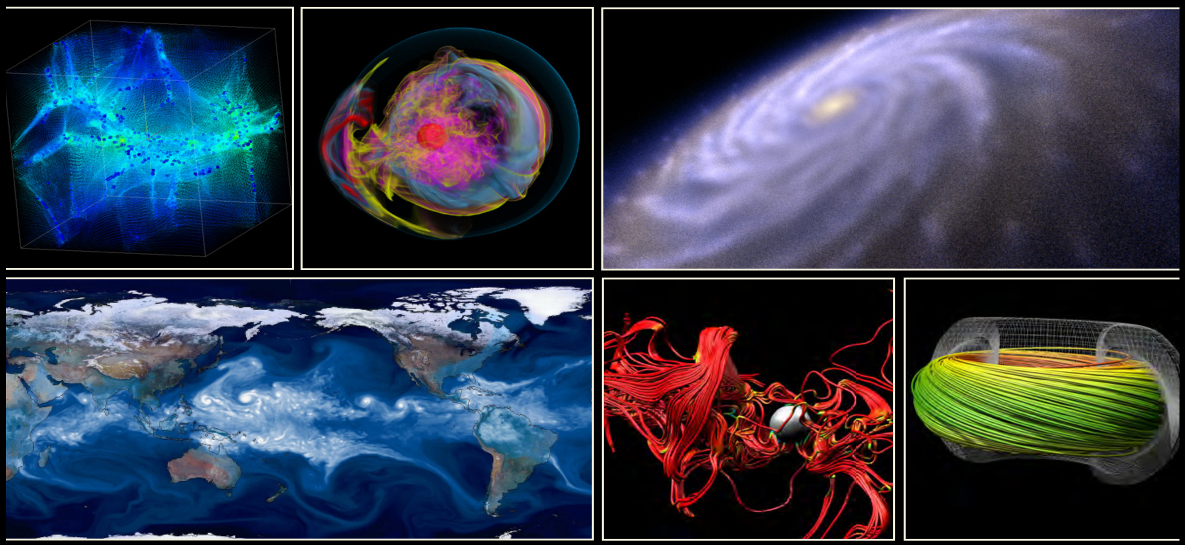
Summary: Exascale Visualization

- Visualization of large data requires techniques for scale and complexity
- Exascale computing will be power constrained and data movement looms large
  - Visualization is unique: we are doing data consumption and the machine is being built for data producers
- In addition to the I/O "wolf", we will now have to deal with a data movement "wolf", plus its 4 pups:
  1) In Situ System
  2) Programming Language
  3) Memory Efficiency
  4) In Situ-Fueled Exploration

SDAV

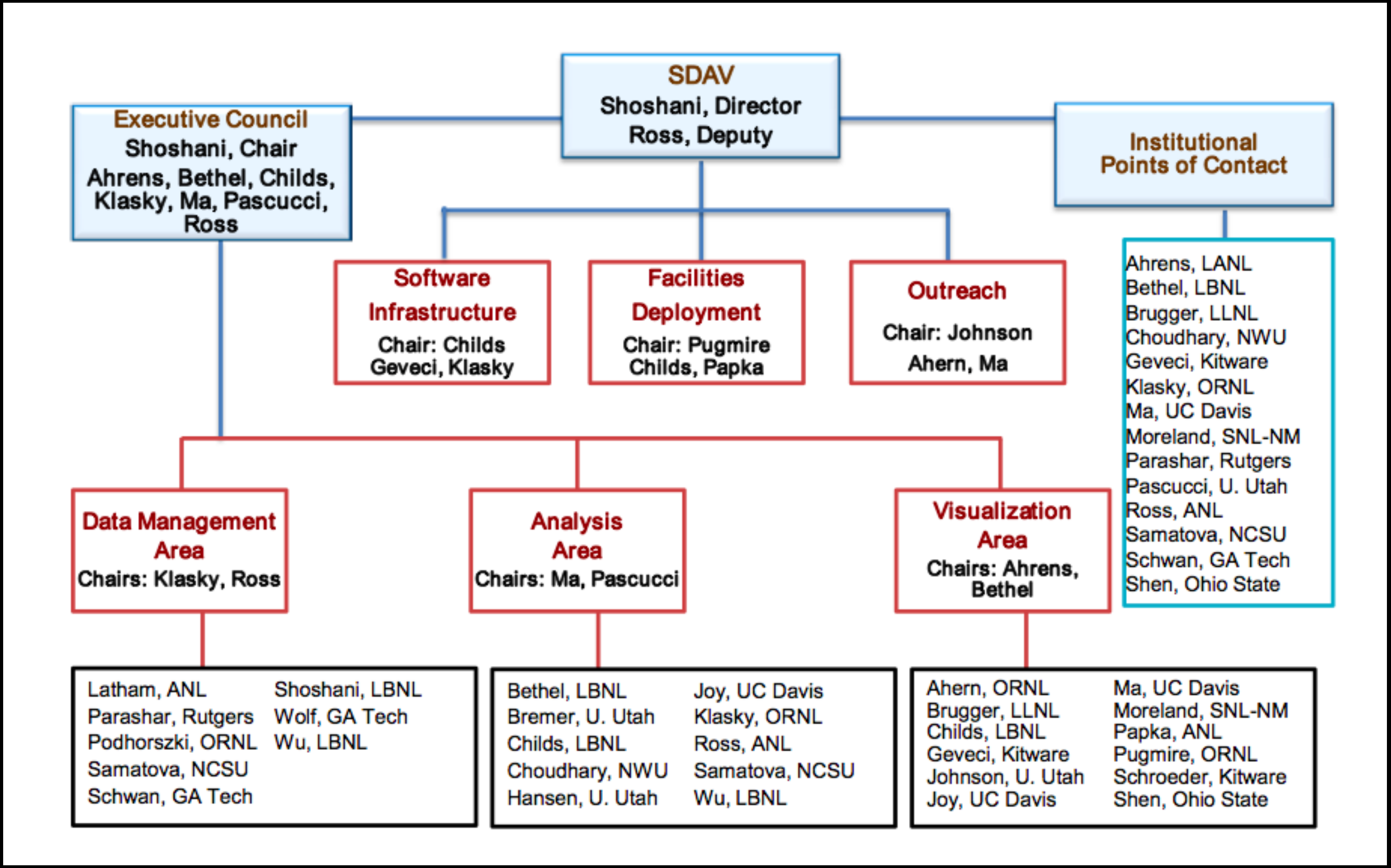
Scalable Data Management, Analysis, and Visualization Institute

● In-situ operation ● Visualization ● Flexible Analysis ● Advanced Architectures ● Scientific Softwar



Lead institution: Lawrence Berkeley National Laboratory Berkeley CA, 94720

SDAV Institute Management Structure



SDAV
Shoshani, Director
Ross, Deputy

Executive Council
Shoshani, Chair
Ahrens, Bethel, Childs, Klasky, Ma, Pascucci, Ross

Institutional Points of Contact

Ahrens, LANL
Bethel, LBNL
Brugger, LLNL
Choudhary, NWU
Geveci, Kitware
Klasky, ORNL
Ma, UC Davis
Moreland, SNL-NM
Parashar, Rutgers
Pascucci, U. Utah
Ross, ANL
Samatova, NCSU
Schwan, GA Tech
Shen, Ohio State

Software Infrastructure
Chair: Childs
Geveci, Klasky

Facilities Deployment
Chair: Pugmire
Childs, Papka

Outreach
Chair: Johnson
Ahern, Ma

Data Management Area
Chairs: Klasky, Ross

Latham, ANL
Parashar, Rutgers
Podhorszki, ORNL
Samatova, NCSU
Schwan, GA Tech
Shoshani, LBNL
Wolf, GA Tech
Wu, LBNL

Analysis Area
Chairs: Ma, Pascucci

Bethel, LBNL
Bremer, U. Utah
Childs, LBNL
Choudhary, NWU
Hansen, U. Utah
Joy, UC Davis
Klasky, ORNL
Ross, ANL
Samatova, NCSU
Wu, LBNL

Visualization Area
Chairs: Ahrens, Bethel

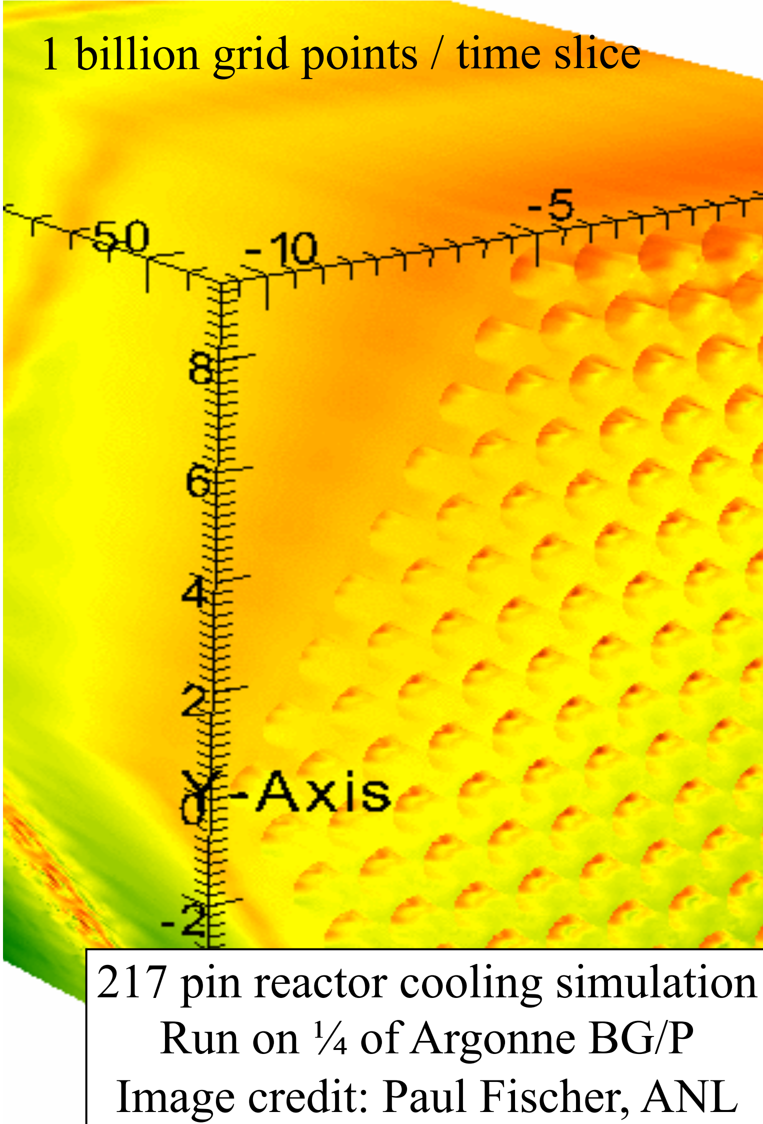
Ahern, ORNL
Brugger, LLNL
Childs, LBNL
Geveci, Kitware
Johnson, U. Utah
Joy, UC Davis
Ma, UC Davis
Moreland, SNL-NM
Papka, ANL
Pugmire, ORNL
Schroeder, Kitware
Shen, Ohio State

Goal

- The goal of SDAV is twofold:
  - to actively work with application teams to assist them in achieving breakthrough science;
  - to provide technical solutions in the data management, analysis, and visualization regimes that are broadly used by the computational science community.

VisIt is an open source, richly featured, turn-key application for large data.

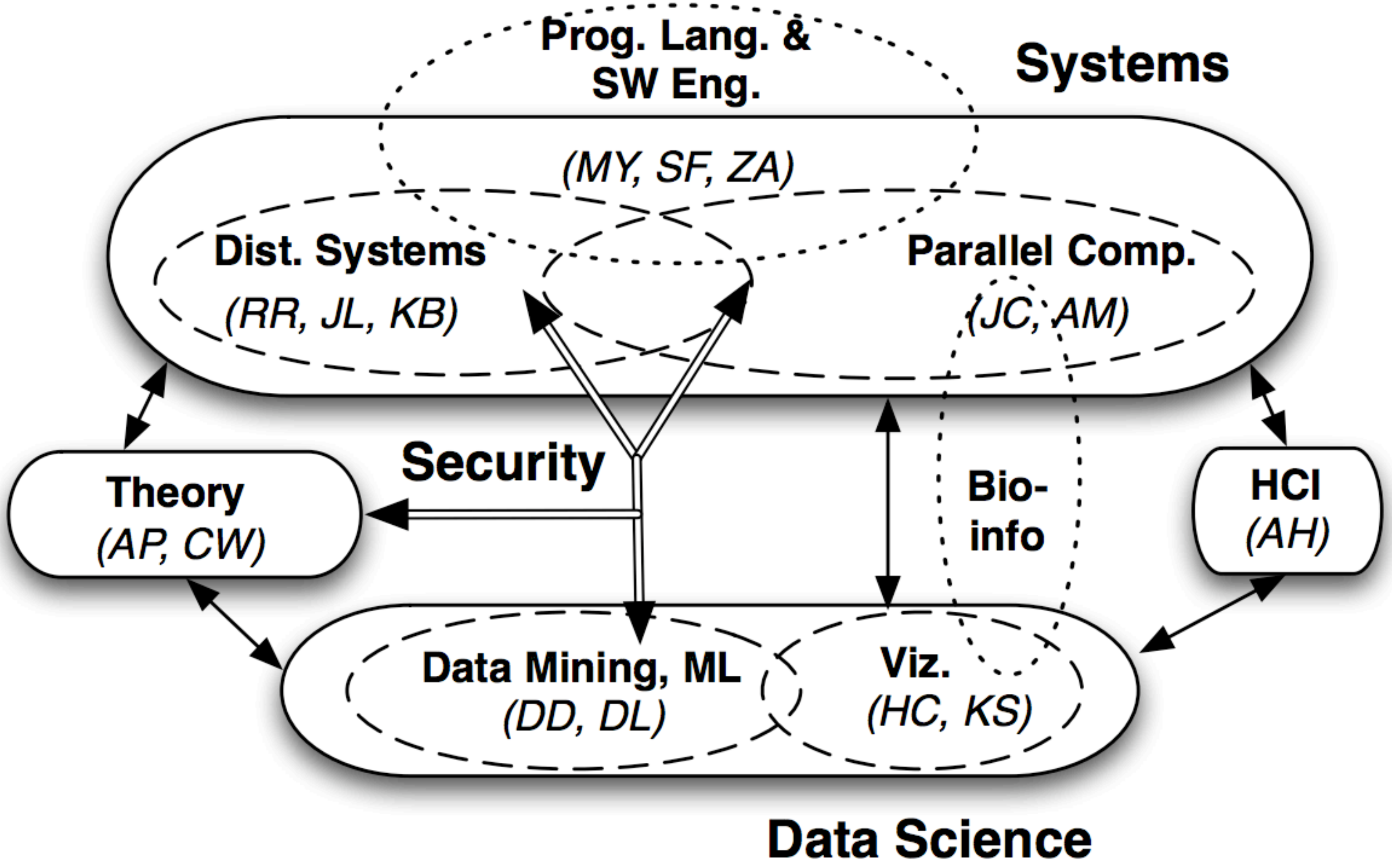
- For data exploration, quantitative analysis, communication, debugging, & more.
- >400 filters
- ~15 active developers
- Popular
  - R&D 100 award in 2005
  - Used on many of the Top500
  - >200K downloads
  - Funded by DOE/NNSA, DOE/NE, DOE/ASCR, NSF/XSEDE, & more



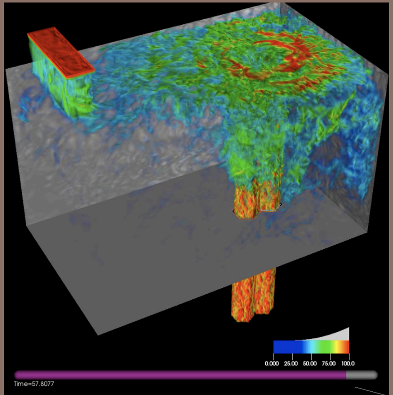
1 billion grid points / time slice

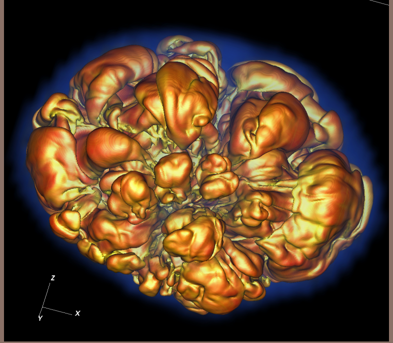
217 pin reactor cooling simulation
Run on ¼ of Argonne BG/P
Image credit: Paul Fischer, ANL

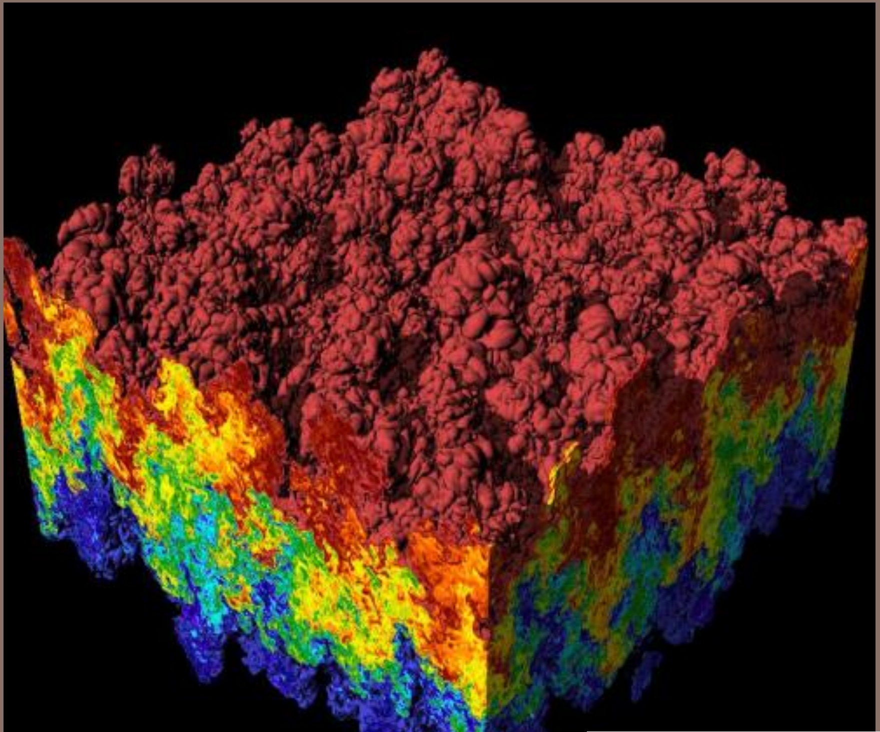
University of Oregon CIS Department

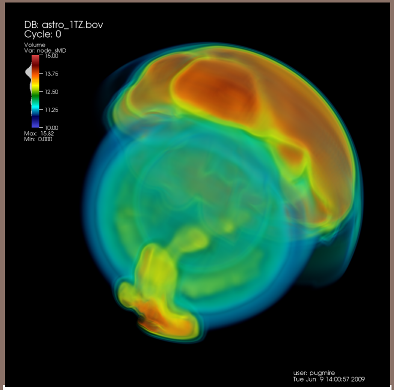


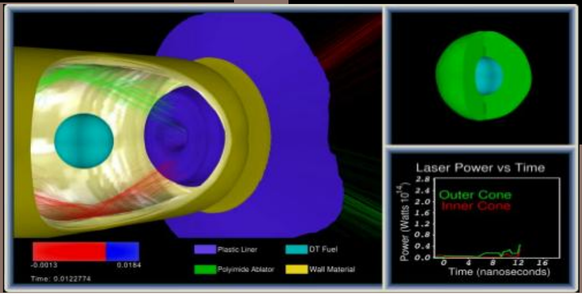
EXASCALE VISUALIZATION: GET READY FOR A WHOLE NEW WORLD

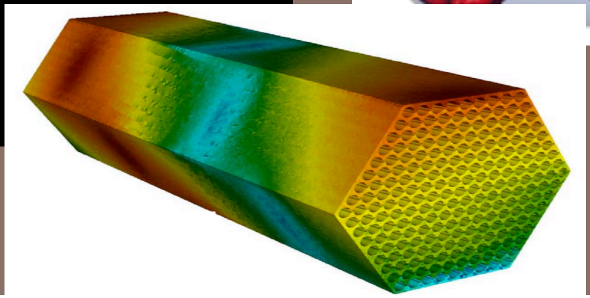












March 26, 2013

Hank Childs, University of Oregon